

An Equivalence Preserving Transformation from the Fibonacci to the Galois NLFSRs

Elena Dubrova

Royal Institute of Technology (KTH), Electrum 229, 164 46 Kista, Sweden
dubrova@kth.se

Abstract. Conventional Non-Linear Feedback Shift Registers (NLFSRs) use the Fibonacci configuration in which the value of the first bit is updated according to some non-linear feedback function of previous values of other bits, and each remaining bit repeats the value of its previous bit. We show how to transform the feedback function of a Fibonacci NLFSR into several smaller feedback functions of individual bits. Such a transformation reduces the propagation time, thus increasing the speed of pseudo-random sequence generation. The practical significance of the presented technique is that it makes possible increasing the keystream generation speed of any Fibonacci NLFSR-based stream cipher with no penalty in area.

Keywords: Fibonacci NLFSR, Galois NLFSR, pseudo-random sequence, keystream, stream cipher.

1 Introduction

Non-Linear Feedback Shift Registers (NLFSRs) have been proposed as an alternative to Linear Feedback Shift Registers (LFSRs) for generating pseudo-random sequences for stream ciphers. NLFSR-based stream ciphers include *Achterbahn* [1], *Dragon* [2], *Grain* [3], *Trivium* [4], *VEST* [5], and [6]. NLFSRs have been shown to be more resistant to cryptanalytic attacks than LFSRs [7,8]. However, construction of large NLFSRs with guaranteed long periods remains an open problem. A systematic algorithm for NLFSR synthesis has not been discovered so far. Only some special cases have been considered [9,10,11,12,13,14,15,16,17].

In general, there are two ways to implement an NLFSR: in the Fibonacci configuration, or in the Galois configuration. The *Fibonacci* configuration, shown in Figure 1, is conceptually more simple. The Fibonacci type of NLFSRs consists of a number of bits numbered from left to right as $n-1, n-2, \dots, 0$ with feedback from each bit to the $n-1$ th bit. At each clocking instance, the value of the bit i is moved to the bit $i-1$. The value of the bit 0 becomes the output of the register. The new value of the bit $n-1$ is computed as some non-linear function of the previous values of other bits.

In the *Galois* type of NLFSR, shown in Figure 2, each bit i is updated according to its own feedback function. Thus, in contrast to the Fibonacci NLFSRs in which feedback is applied to the $n-1$ th bit only, in the Galois NLFSRs feedback is potentially applied to every bit. Since the next state functions of individual bits of a Galois NLFSR

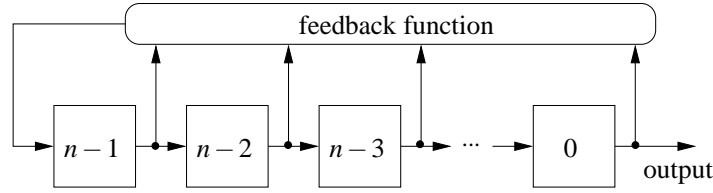


Fig. 1. An Fibonacci type of NLFSR.

are computed in parallel, the propagation time is reduced to that of smaller functions of individual bits. This makes Galois NLFSRs particularly attractive for stream ciphers application in which high keystream generation speed is important.

However, Galois NLFSRs also have the following two drawbacks:

1. An n -bit Galois NLFSR with the period of $2^n - 1$ does not necessarily satisfy the 1st and the 2nd postulates of Golomb [18]. An n -bit Fibonacci NLFSR with the period of $2^n - 1$ always satisfy both postulates [9].
2. The period of the output sequence of a Galois NLFSR is not necessarily equal to the length of the longest cyclic sequence of its consecutive states [18]. The period of a Fibonacci NLFSR always equals to the longest cyclic sequence of its consecutive states [9].

These drawbacks do not create any problems in the linear case because, for LFSRs, there exist a one-to-one mapping between the Fibonacci and Galois configurations. A Galois LFSR generating the same output sequence as a given Fibonacci LFSR (and therefore possessing none of the above mentioned drawbacks) can be obtained by reversing the order of the feedback taps and adjusting the initial state. For example, Figure 3 shows the Fibonacci and Galois configurations for the generator polynomial $x^3 + x + 1$. If the Fibonacci LFSR is initialized to the state 001 and the Galois one is initialized to the state 101, then they generate the same periodic sequence 1001011.

In the non-linear case, however, no mapping between the Fibonacci and the Galois configurations has been known until now. The problem of finding such a mapping is addressed in this paper. We show that, for each Fibonacci NLFSR, there exist a class of equivalent Galois NLFSRs which produce the same output sequence. We show how to transform a given Fibonacci NLFSR into an equivalent Galois NLFSR.

The most significant contribution of the paper is a sufficient condition for equivalence of two NLFSRs before and after the transformation. It is formulated and proved for the general case which covers not only the equivalence between a Fibonacci and a Galois NLFSRs, but only the equivalence between two Galois NLFSRs.

The paper is organized as follows. Section 2 describes main notions and definitions used in the sequel. Section 3 formulates a sufficient condition for existence of a non-linear recurrence describing the output sequence of an NLFSR. Section 4 presents a sufficient condition for the equivalence of two NLFSRs. In Section 5, we define a Galois NLFSR which is unique for a given Fibonacci NLFSR and show how to compute it. Section 6 concludes the paper and discusses open problems.

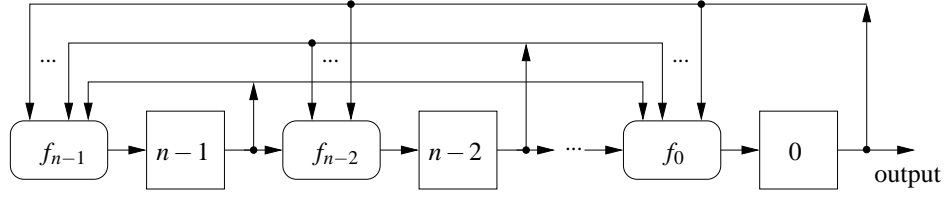


Fig. 2. A Galois type of NLFSR.

2 Preliminaries

In this section, we describe basic definitions and notation used in the sequel.

The *algebraic normal form (ANF)* of a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a polynomial in $GF(2)$ of type

$$f(x_0, \dots, x_{n-1}) = \sum_{i=0}^{2^n-1} c_i \cdot x_0^{i_0} \cdot x_1^{i_1} \cdot \dots \cdot x_{n-1}^{i_{n-1}},$$

where $c_i \in \{0, 1\}$ and $(i_0 i_1 \dots i_{n-1})$ is the binary expansion of i with i_{n-1} being the least significant bit.

The *dependence set* (or *support set*) of a Boolean function f is defined by

$$dep(f) = \{i \mid f|_{x_i=0} \neq f|_{x_i=1}\},$$

where $f|_{x_i=j} = f(x_0, \dots, x_{i-1}, j, x_{i+1}, \dots, x_{n-1})$ for $j \in \{0, 1\}$.

Let $\alpha_{min}(f)$ ($\alpha_{max}(f)$) be the smallest (largest) index of variables in $dep(f)$.

Let $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$ be a feedback function of the bit i , $i \in \{0, 1, \dots, n-1\}$, of an NLFSR. All results in this paper as derived for NLFSRs whose feedback functions are *singular* functions of type

$$f_i = x_{i+1} \oplus g_i(x_0, \dots, x_{n-1}), \quad (1)$$

where $g_i : \{0, 1\}^{n-1} \rightarrow \{0, 1\}$, $i+1 \notin dep(g_i)$, and the sign “+” is modulo n . Singularity guarantees that the state transition graph of an NLFSR is “branchless”, i.e. that each state belongs to one of the state cycles [9].

Let $s_i(t)$ denote the value of the bit i at time t . The sequence of states an n -bit NLFSR with the singular feedback functions can be described by a system of n non-linear equations of type:

$$\begin{cases} s_{n-1}(t) = s_0(t-1) \oplus g_{n-1}(s_1(t-1), s_2(t-1), \dots, s_{n-1}(t-1)) \\ s_{n-2}(t) = s_{n-1}(t-1) \oplus g_{n-2}(s_0(t-1), s_1(t-1), \dots, s_{n-2}(t-1)) \\ \dots \\ s_0(t) = s_1(t-1) \oplus g_0(s_0(t-1), s_2(t-1), \dots, s_{n-1}(t-1)). \end{cases} \quad (2)$$

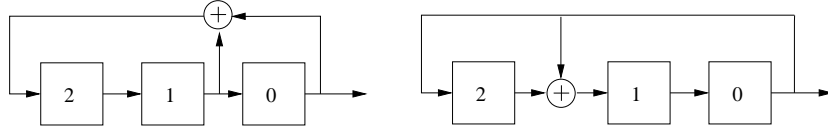


Fig. 3. The Fibonacci LFSR (left) and the Galois LFSR (right) for the generator polynomial $x^3 + x + 1$.

3 A Condition for Existence of a Non-Linear Recurrence

In this section, we formulate a condition for existence of a non-linear recurrence describing the output sequence of an NLFSR. First, we introduce some definitions which are necessary for the presentation of main results.

Definition 1. Two NLFSRs are equivalent if there are initial states, possibly different for each NLFSR, from which they generate the same output sequences.

Definition 2. The feedback graph of an NLFSR has n vertices v_0, \dots, v_{n-1} representing the bits $0, \dots, n-1$. There is an edge from v_i to v_j if $i \in \text{dep}(f_j)$, $i, j \in \{0, 1, \dots, n-1\}$.

Definition 3. The terminal bit of an n -bit NLFSR is the bit with the largest index i which satisfies the following condition: For all bits j such that $i > j \geq 0$, the feedback function f_j is of type $f_j = x_{j+1}$, $i, j \in \{0, 1, \dots, n-1\}$.

Definition 4. The operation substitution, denoted by $\text{sub}(v_i, v_j)$, is defined for any vertex v_i which has a unique predecessor v_j . The substitution $\text{sub}(v_i, v_j)$ removes v_i from the feedback graph and, for each successor v_k of v_i , replaces the edge (v_i, v_k) by an edge (v_j, v_k) , $i, j, k \in \{0, \dots, n-1\}$.

Definition 5. Given a feedback graph G , the reduced feedback graph of G is a graph obtained by subsequently applying the substitution to all vertices of G with the input degree 1.

Since substitution merges a vertex with its unique predecessor, the order of applying the substitution does not influence the resulting reduced feedback graph, i.e. it is unique for a given G .

Lemma 1. If the feedback graph of an n -bit NLFSR can be reduced to a single vertex v_i , $i \in \{0, 1, \dots, n-1\}$, then there exist a non-linear recurrence describing the sequence of values of the bit i of type

$$s_i(t) = \sum_{j=0}^{2^n-1} (a_j \cdot \prod_{k=0}^{n-1} s^{j_k}(t-n+k)), \quad (3)$$

where $a_j \in \{0, 1\}$, $(j_0 j_1 \dots j_{n-1})$ is the binary expansion of j with j_{n-1} being the least significant bit, and $s^{j_k}(t-n+k)$ is defined as follows

$$s^{j_k}(t-n+k) = \begin{cases} s(t-n+k), & \text{for } i = 1, \\ 1, & \text{for } i = 0. \end{cases}$$

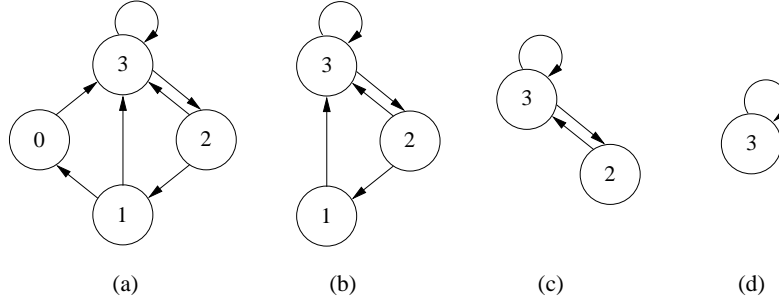


Fig. 4. Reduction steps for the feedback graph of the Fibonacci NLFSR from the example: (a) initial graph; (b) after $sub(v_0, v_1)$; (c) after $sub(v_1, v_2)$; (d) after $sub(v_2, v_3)$.

Proof: Let v_i be a vertex of the feedback graph which has a unique predecessor v_j and m successors v_{k_1}, \dots, v_{k_m} , $j, k_p \in \{0, 1, \dots, n-1\}$, $p \in \{0, 1, \dots, m\}$. By Df. 2, this implies that $s_i(t) = s_j(t-1)$ and, for each p , $s_{k_p}(t)$ depends on $s_i(t-1)$.

The substitution $sub(v_i, v_j)$ is equivalent to replacing the variable $s_i(t-1)$ in the equation of each $s_{k_p}(t)$ by $s_j(t-2)$. This reduces the number of variables in the equations (2) by one and reduces the number of equations by one.

If the feedback graph of an NLFSR can be reduced to a single vertex, say v_r , then the substitution can be applied $n-1$ times. So, the number of variables in the equations (2) can be reduced to a single variable and the number of equations can be reduced to a single equation. This equation corresponds to the non-linear recurrence relation describing the sequence of states of the bit r of the NLFSR.

□

Example 1: As an example, consider a 4-bit Fibonacci NLFSR with the feedback function $f_3 = x_0 \oplus x_1 \oplus x_2 \oplus x_1 x_3$. Its sequence of states can be described by the following equations:

$$\begin{cases} s_3(t) = s_0(t-1) \oplus s_1(t-1) \oplus s_2(t-1) \oplus s_1(t-1)s_3(t-1), \\ s_2(t) = s_3(t-1), \\ s_1(t) = s_2(t-1), \\ s_0(t) = s_1(t-1). \end{cases}$$

This NLFSR generates the following output sequence with the period 15:

111011000101001...

The feedback graph of this NLFSR is shown in Figure 4(a). It can be reduced to a single vertex as follows:

1. $sub(v_0, v_1)$ reduces the graph to Figure 4(b). This is equivalent to substituting $s_0(t)$ by $s_1(t-1)$ into the equation of $s_3(t)$:

$$s_3(t) = s_1(t-2) \oplus s_1(t-1) \oplus s_2(t-1) \oplus s_1(t-1)s_3(t-1).$$

2. $sub(v_1, v_2)$ reduces the graph to Figure 4(c). This is equivalent to substituting $s_1(t)$ by $s_2(t-1)$ into the equation of $s_3(t)$:

$$s_3(t) = s_2(t-3) \oplus s_2(t-2) \oplus s_2(t-1) \oplus s_2(t-2)s_3(t-1).$$

3. $sub(v_2, v_3)$ reduces the graph to Figure 4(d). This is equivalent to substituting $s_2(t)$ by $s_3(t-1)$ into the equation of $s_3(t)$:

$$s_3(t) = s_3(t-4) \oplus s_3(t-3) \oplus s_3(t-2) \oplus s_3(t-3)s_3(t-1).$$

This gives us a non-linear recurrence describing the sequence of values of the bit 3. Since other bits repeat the content of the 3rd bit, the recurrence is identical for all bits, and thus for the output of the NLFSR.

It is easy to see that the feedback graph of a Fibonacci NLFSR can always be reduced to a single vertex v_{n-1} . Therefore, for a Fibonacci NLFSR, a non-linear recurrence of type (3) always exists. Its coefficients a_i , $i \in \{0, 1, \dots, 2^n - 1\}$, are equal to the coefficients c_i of the ANF of the feedback function f_{n-1} .

For Galois NLFSRs, a non-linear recurrence of type (3) may or may not exist. If it exists, it may be different for different bits.

Example 2: As another example, consider a Galois NLFSR with the following feedback functions:

$$\begin{aligned} f_3 &= x_0 \oplus x_1 x_3, \\ f_2 &= x_3, \\ f_1 &= x_2, \\ f_0 &= x_1 \oplus x_2 \oplus x_3. \end{aligned}$$

Its feedback graph can be reduced to the vertex v_3 , giving us the following recurrence:

$$s_3(t) = s_3(t-4) \oplus s_3(t-3) \oplus s_3(t-2) \oplus s_3(t-3)s_3(t-1).$$

This recurrence is the same as the one of the Fibonacci NLFSR from the Example 1. Bits 2 and 1 repeat the same recurrence as the bit 3, however, the value of the bit 0 is the XOR of the bits 1, 2 and 3. Thus, its sequence of values differs from the one of the 3rd bit. Therefore, the output sequence of this Galois NLFSR, is different the output sequence of the Fibonacci NLFSR from the Example 1.

4 A Transformation from the Fibonacci to the Galois NLFSRs

In this section, we show how to transform a Fibonacci NLFSR into an equivalent Galois NLFSR.

Let P_f denote the set of all product-terms of the ANF of a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Given an ANF product-term $p \in P_f$, the notation p_{-k} means that the index of each variable x_i of p is changed to x_{i-k} , where “ $-$ ” is modulo n .

For example, if $n = 4$, and $p = x_0 x_1 x_3$ then

$$p_{-1} = x_3 x_0 x_2, \quad p_{-2} = x_2 x_3 x_1, \quad p_{-3} = x_1 x_2 x_0.$$

Definition 6. The operation shifting, denoted by $f_a \xrightarrow{p} f_b$, $p \in P_{f_a}$, $a, b \in \{0, 1, \dots, n-1\}$, $b < a$, removes the product-term p from the ANF of the function f_a and adds the product-term $p_{-(a-b)}$ to the ANF of the function f_b .

As we can see from the definition, shifting subtracts $(a - b)$ from the index of each variable in the shifted product-term (modulo n). For example, if initially

$$\begin{aligned} f_3 &= x_0 \oplus x_1 x_3 \\ f_2 &= x_3 \end{aligned}$$

then, after $f_3 \xrightarrow{x_1 x_3} f_2$, we get

$$\begin{aligned} f_3 &= x_0 \\ f_2 &= x_3 \oplus x_0 x_2. \end{aligned}$$

Definition 7. An n -bit NLFSR is uniform if:

- (a) all its feedback functions are of type (1), and
- (b) for all its bits i such that $n - 1 \geq i > \tau$, the following condition holds:

$$\alpha_{\max}(g_i) \leq \tau, \quad (4)$$

where τ is the terminal bit of the NLFSR, $\tau \in \{0, 1, \dots, n-1\}$.

Note that any Fibonacci NLFSR is uniform.

Lemma 2. If an NLFSR is uniform, then its feedback graph can be reduced to a single vertex.

Proof: Suppose that an NLFSR N is uniform. We show that then we can always reduce the feedback graph of N to the vertex v_τ corresponding to the terminal bit τ of N .

By Df. 3, for $i \in \{0, 1, \dots, \tau-1\}$, each vertex v_i of the feedback graph has input degree 1. So, for each $i \in \{0, 1, \dots, \tau-1\}$, we can apply the substitution $\text{sup}(v_i, v_{i+1})$ to remove v_i from the feedback graph, and, for each successor v_k of v_i , to replace the edge (v_i, v_k) by an edge (v_τ, v_k) . Therefore, by applying a sequence of substitutions $\text{sup}(v_0, v_1), \text{sup}(v_1, v_2), \dots, \text{sup}(v_{\tau-1}, v_\tau)$ we can remove $v_0, v_1, \dots, v_{\tau-1}$ from the feedback graph and change the origin of all outgoing edges of $v_0, v_1, \dots, v_{\tau-1}$ to v_τ .

Since the condition (4) holds and the origin of all outgoing edges of $v_0, v_1, \dots, v_{\tau-1}$ is changed to v_τ , each of the vertices v_i for $i \in \{\tau+1, \tau+2, \dots, n-1\}$ has no more than two incoming edges: one from v_{i+1} and one from v_τ . This implies that each of them has the output degree 1.

Clearly, v_{n-1} has only one incoming edge, from v_τ . By applying the substitution $\text{sup}(v_{n-1}, v_\tau)$, we can remove v_{n-1} and replace the edge (v_{n-1}, v_{n-2}) by the edge (v_τ, v_{n-2}) . This makes the input degree of v_{n-2} one. Continuing similarly with the sequence of substitutions $\text{sup}(v_{n-2}, v_\tau), \dots, \text{sup}(v_{\tau+1}, v_\tau)$ we remove $v_{n-2}, \dots, v_{\tau+1}$ and reduce the graph to one vertex, v_τ .

□

The above condition is sufficient, but not necessary. For example, the NLFSR from the Example 2 is not uniform, but it can be reduced to a single vertex.

The following theorem is the main result of the paper. It presents a sufficient condition for equivalence of two NLFSRs. Note, that it is formulated for shiftings on sub-functions g_i of the singular feedback functions f_i (see the expression 1), because the variable x_{i+1} should not be shifted in order to preserve the register structure.

Theorem 1. *Given a uniform NLFSR, a shifting $g_a \xrightarrow{p} g_b$, $a, b \in \{0, 1, \dots, n-1\}$, $b < a$, $P \subseteq P_{g_a}$, preserves the equivalence if the transformed NLFSR is uniform as well.*

Proof: See Appendix.

The condition of the Theorem 1 is sufficient, but not necessary. For example, the following NLFSR can be obtained from the NLFSR from the Example 1 by applying the shifting $f_3 \xrightarrow{x_1 x_3} f_0$, $f_3 \xrightarrow{x_1} f_1$ and $f_3 \xrightarrow{x_2} f_1$:

$$\begin{aligned} f_3 &= x_0, \\ f_2 &= x_3, \\ f_1 &= x_2 \oplus x_0 \oplus x_3, \\ f_0 &= x_1 \oplus x_0 x_2. \end{aligned}$$

This NLFSR is not uniform, however, it is equivalent to the NLFSR from the Example 1.

Next, we formulate a condition which should be satisfied in order to obtain a uniform NLFSR after shifting.

Theorem 2. *Given a uniform NLFSR N , an NLFSR obtained from N by a shifting $g_a \xrightarrow{p} g_b$, $a, b \in \{0, 1, \dots, n-1\}$, $b < a$, $P \subseteq P_{g_a}$, is uniform only if*

$$b \geq a - \alpha_{\min}(p). \quad (5)$$

Proof: If $b < a - \alpha_{\min}(p)$, then $\alpha_{\min}(p) < a - b$. Therefore, after the shifting $g_a \xrightarrow{p} g_b$, $\alpha_{\min}(p)$ becomes $\alpha_{\min}(p) + n - (a - b) = \alpha_{\min}(p) + b + (n - a)$. By Df. 6, $b < a$, thus a is always greater than 0. So, for any $a \in \{1, 2, \dots, n-1\}$, after shifting the feedback function g_b contains a product-term whose index is greater than b by $n - a$. Since the terminal bit of the NLFSR is smaller or equal to b , the condition (4) of Df. 7 is violated.

□

Often an equivalent Galois NLFSR can be obtained from a Fibonacci NLFSR by shifting product-terms one-by-one. Sometimes, however, more than one product-term has to be shifted in order to preserve the equivalence. For example, if the feedback function g_{n-1} has more than one product-term containing the variable x_{n-1} , then all such product-terms have to be shifted. The Lemma below describes two cases in which the product-terms can be shifted one-by-one.

Lemma 3. *Given a uniform NLFSR with the terminal bit τ and a shifting $g_a \xrightarrow{p} g_b$, $a, b \in \{0, 1, \dots, n-1\}$, $b < a$, $P \subseteq P_{g_a}$, the following holds:*

- (a) If $b \geq \tau$, then $g_a \xrightarrow{p} g_b$ preserves the equivalence for any $p \in P_{g_a}$ which satisfies the condition (5).
- (b) If $b < \tau$ and $\alpha_{\max}(g_i) \leq b$ for all $i \in \{n-1, n-2, \dots, b\}$, then $g_a \xrightarrow{p} g_b$ preserves the equivalence for any $p \in P_{g_a}$ which satisfies the condition (5).

Proof: Case (a): By Df. 6, after the shifting $\alpha_{\min}(p)$ becomes $\alpha_{\min}(p) - (a - b)$. Since the condition (5) is satisfied, $\alpha_{\min}(p) \geq a - b$, i.e. after shifting the indexes of variables of p are reduced by some value between 1 and $\alpha_{\min}(p)$. Therefore, after the shifting, none of the product-terms of p violates the condition (4). Since the initial NLFSR is uniform and the terminal bit is not changed, the transformed NLFSR is uniform as well, and therefore, by Theorem 1, the equivalence is preserved.

Case (b): Similarly to the case (a) we can show that, after the shifting, none of the product-terms of p violates the condition (4). Since $\alpha_{\max}(g_i) \leq b$ for all i by assumption, the transformed NLFSR is uniform and therefore, by Theorem 1, the equivalence is preserved. □

The above Lemma implies that, for any Fibonacci NLFSR, shifting can always reduce the index of the initial terminal bit $n - 1$ at least by 1. It reduces the index of the terminal bit exactly by 1 if g_{n-1} of the Fibonacci NLFSR contains a product with $\alpha_{\max}(g_i) = n - 1$ and $\alpha_{\min}(g_{n-1}) = 1$. The smaller the difference between $\alpha_{\max}(g_{n-1})$ and $\alpha_{\min}(g_{n-1})$, the more the index of the initial terminal bit can be reduced.

5 Fully Shifted Galois NLFSRs

Usually, there are multiple ways to transform a Fibonacci NLFSR into a Galois NLFSR. Next, we define a “fully shifted” Galois NLFSR which is unique for a given Fibonacci NLFSR and show how to compute it.

Definition 8. An NLFSR is fully shifted if no product-term of any function g_i can be shifted to a function g_j with the index $j < i$ without violating the condition (4), $i, j \in \{0, 1, \dots, n - 1\}$.

In the linear case, a fully shifted NLFSR reduces to a Galois LFSR, i.e. it is a generalization of the Galois LFSR. Note that this is not the case for NLFSRs which are not fully shifted.

Algorithm 1: Given a uniform n -bit Fibonacci NLFSR N , the fully shifted Galois NLFSR \hat{N} which is equivalent to N is obtained as follows.

First, the terminal bit τ of \hat{N} is computed as:

$$\tau = \max_{\substack{\forall p \in P_{g_{n-1}} \\ \text{with } |p| > 1}} (\alpha_{\max}(p) - \alpha_{\min}(p)), \quad (6)$$

where $|p|$ denotes the number of variables in the product-term p .

Then, each product-term $p \in P_{g_{n-1}}$ with $\alpha_{\min}(p) \leq (n-1) - \tau$ is shifted to $g_{n-1-\alpha_{\min}(p)}$:

$$g_{n-1} \xrightarrow{p} g_{n-1-\alpha_{\min}(p)}.$$

and each product-term $p \in P_{g_{n-1}}$ with $\alpha_{\min}(p) > (n-1) - \tau$ is shifted to g_τ :

$$g_{n-1} \xrightarrow{p} g_\tau.$$

Theorem 3. *Algorithm 1 correctly computes the fully shifted Galois NLFSR for a given Fibonacci NLFSR.*

Proof: For each product p such that $\alpha_{\min}(p) \leq (n-1) - \tau$, the indexes are reduced by $\alpha_{\min}(p)$. So, after the shifting, the smallest index becomes 0 and the largest becomes $\alpha_{\max}(p) - \alpha_{\min}(p)$. By equation (6), $\alpha_{\max}(p) - \alpha_{\min}(p) \leq \tau$.

For each product p such that $\alpha_{\min}(p) > (n-1) - \tau$, the indexes are reduced by $(n-1) - \tau$. Since $\alpha_{\min}(p) < \alpha_{\max}(p) \leq n-1$, the largest index after the shifting is $0 < \alpha_{\max}(p) - ((n-1) - \tau) \leq \tau$. Since $(n-1) - \tau < \alpha_{\min}(p) < \alpha_{\max}(p)$, the smallest index after the shifting is $0 < \alpha_{\min}(p) - ((n-1) - \tau) < \tau$.

So, the transformed NLFSR \hat{N} is uniform and therefore, by Theorem 1, two NLFSRs are equivalent. It remains to prove that \hat{N} is fully shifted.

By Df 6, index of each variable of p is reduced by $\alpha_{\min}(p)$ after the shifting. Therefore, for each product-term $p \in P_{g_{n-1}}$ such that $\alpha_{\min}(p) \leq \tau$, p after the shifting contains a variable x_0 . If p is shifted further from $g_{n-1-\alpha_{\min}(p)}$ to $g_{n-1-\alpha_{\min}(p)-i}$ for some $1 \leq i \leq n-1 - \alpha_{\min}(p)$, the index of x_0 increases to $n-i$. For every value of i in the range $1 \leq i \leq n-1 - \alpha_{\min}(p)$, $n-i > n-1 - \alpha_{\min}(p)$, so the condition (4) is violated and the resulting NLFSR is not equivalent to the initial Fibonacci NLFSR.

Each product-term $p \in P_{g_{n-1}}$ such that $\alpha_{\min}(p) > \tau$ is shifted to the terminal bit τ . If p is shifted to some $i < \tau$, then, according to the equation (6), there is a product-term p^* which has $\alpha_{\max}(p^*) > i$ after shifting. Thus, the condition (4) is violated and the resulting NLFSR is not equivalent to the initial Fibonacci NLFSR.

□

Example 4: As an example, consider the following 32-bit Fibonacci NLFSR which is used in the NLFSR-based stream cipher from [6]:

$$f_{31} = x_0 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus x_{17} \oplus x_{20} \oplus x_{27} \oplus x_{30} \oplus x_3 x_9 \oplus x_{12} x_{15} \oplus x_4 x_5 x_{16}$$

Its corresponding fully shifted Galois NLFSR has the terminal bit $\tau = 12$ and the following feedback functions:

$$\begin{aligned} f_{31} &= x_0 \\ f_{29} &= x_{30} \oplus x_0 \\ f_{28} &= x_{29} \oplus x_0 x_6 \\ f_{27} &= x_{28} \oplus x_0 x_{12} \\ f_{25} &= x_{26} \oplus x_0 \\ f_{24} &= x_{25} \oplus x_0 \\ f_{19} &= x_{20} \oplus x_0 \oplus x_0 x_3 \\ f_{14} &= x_{15} \oplus x_0 \\ f_{12} &= x_{13} \oplus x_1 \oplus x_8 \oplus x_{11} \end{aligned}$$

The functions which are omitted are of type $f_i = f_{i+1}$. This NLFSR has 7 feedback variables: $x_0, x_1, x_3, x_6, x_8, x_{11}$ and x_{12} , while the Fibonacci NLFSR has 15 feedback variables.

We can further reduce the depth of circuits implementing feedback functions and the number of feedback variables as follows:

$$\begin{aligned}
f_{31} &= x_0 \\
f_{29} &= x_{30} \oplus x_0 \\
f_{28} &= x_{29} \oplus x_0 x_6 \\
f_{27} &= x_{28} \oplus x_0 x_1 x_{12} \\
f_{25} &= x_{26} \oplus x_0 \\
f_{24} &= x_{25} \oplus x_0 \\
f_{20} &= x_{21} \oplus x_1 x_4 \\
f_{19} &= x_{20} \oplus x_0 \\
f_{16} &= x_{17} \oplus x_{12} \\
f_{14} &= x_{15} \oplus x_0 \\
f_{13} &= x_{14} \oplus x_{12} \\
f_{12} &= x_{13} \oplus x_1
\end{aligned}$$

This NLFSR has 5 feedback variables: x_0, x_1, x_4, x_6 and x_{12} .

6 Conclusion

In this paper, we show how to transform a Fibonacci NLFSR into the Galois configuration.

The most important open problem is finding an algorithm for constructing NLFSRs with a guaranteed long period. This problem is hard because there seems to be no simple algebraic theory supporting it. Specifically, primitive generator polynomials for LFSR have no analog in the nonlinear case.

References

1. B. Gammel, R. G ttfert, and O. Kniffler, "Achterbahn-128/80: Design and analysis," in *SASC'2007: Workshop Record of The State of the Art of Stream Ciphers*, pp. 152–165, 2007.
2. K. Chen, M. Henricken, W. Millan, J. Fuller, L. Simpson, E. Dawson, H. Lee, and S. Moon, "Dragon: A fast word based stream cipher," in *eSTREM, ECRYPT Stream Cipher Project*, 2005. Report 2005/006.
3. M. Hell, T. Johansson, and W. Meier, "Grain - a stream cipher for constrained environments," citeseer.ist.psu.edu/732342.html.
4. C. D. Canniere and B. Preneel, "TRIVIUM specifications," citeseer.ist.psu.edu/734144.html.
5. B. Gittins, H. A. Landman, S. O'Neil, and R. Kelson, "A presentation on VEST hardware performance, chip area measurements, power consumption estimates and benchmarking in relation to the aes, sha-256 and sha-512." Cryptology ePrint Archive, Report 2005/415, 2005. <http://eprint.iacr.org/>.
6. B. M. Gammel, R. G ttfert, and O. Kniffler, "An NLFSR-based stream cipher," in *ISCAS*, 2006.

7. B. Preneel, "A survey of recent developments in cryptographic algorithms for smart cards," *Comput. Networks*, vol. 51, no. 9, pp. 2223–2233, 2007.
8. A. Canteaut, "Open problems related to algebraic attacks on stream ciphers," in *WCC*, pp. 120–134, 2005.
9. S. Golomb, *Shift Register Sequences*. Aegean Park Press, 1982.
10. J. Mykkeltveit, "Nonlinear recurrences and arithmetic codes," *Information and Control*, vol. 33, no. 3, pp. 193–209, 1977.
11. J. Mykkeltveit, M.-K. Siu, and P. Tong, "On the cycle structure of some nonlinear shift register sequences," *Information and Control*, vol. 43, no. 2, pp. 202–215, 1979.
12. C. A. Ronce, *Feedback Shift Registers*, vol. 169. 1984.
13. C. J. Jansen, *Investigations On Nonlinear Streamcipher Systems: Construction and Evaluation Methods*. Ph.D. Thesis, Technical University of Delft, 1989.
14. M. J. B. Robshaw, *On Binary Sequences with Certain Properties*. Ph.D. Thesis, University of London, 1992.
15. D. Linardatos and N. Kalouptsidis, "Synthesis of minimal cost nonlinear feedback shift registers," *Signal Process.*, vol. 82, no. 2, pp. 157–176, 2002.
16. A. Ahmad, M. J. Al-Mushrafi, and S. Al-Busaidi, "Design and study of a strong cryptosystem model for e-commerce," in *ICCC '02: Proceedings of the 15th international conference on Computer communication*, (Washington, DC, USA), pp. 619–630, International Council for Computer Communication, 2002.
17. J. S. I. Janicka-Lipska, "Boolean feedback functions for full-length nonlinear shift registers," *Telecommunications and Information Technology*, vol. 5, pp. 28–29, 2004.
18. E. Dubrova, M. Teslenko, and H. Tenhunen, "On analysis and synthesis of (n, k) -non-linear feedback shift registers," in *Design and Test in Europe*, 2008. to appear.

7 Appendix: Proof of the Theorem 1

Suppose that the transformed NLFSR is uniform. Then, by Lemma 2, its feedback graph can be reduced to the vertex v_b corresponding to the terminal bit b of the transformed NLFSR after the shifting $g_a \xrightarrow{P} g_b$. So, by Lemma 1, there exists a non-linear recurrence describing the sequence of values of the bit b . It remains to prove that this recurrence is the same as the one of the initial NLFSR.

It is sufficient to consider the case when the shifting $g_a \xrightarrow{P} g_b$ moves a product-term of type $x_k x_a$ for some $k < a$. For product-terms with more variables or the product-term without x_a the proof is similar.

If the shifted product is $x_k x_a$, then the function g_a can be represented as $g_a = g_a^* \oplus x_k x_a$, where $g_a^* = g_a \oplus x_k x_a$. So, the NLFSR before the shifting can be represented by the following system of equations:

$$\begin{cases} s_{n-1}(t) = s_0(t-1) \oplus g_{n-1}(s_0(t-1), s_1(t-1), \dots, s_b(t-1)) \\ \dots \\ s_a(t) = s_{a+1}(t-1) \oplus g_a^*(s_0(t-1), s_1(t-1), \dots, s_b(t-1)) \oplus s_k(t-1)s_a(t-1) \\ s_{a-1}(t) = s_a(t-1) \\ \dots \\ s_0(t) = s_1(t-1) \end{cases}$$

Since $i+1 \notin \text{dep}(g_i)$ for $i \in \{0, 1, \dots, n-1\}$, each g_i does not depends of $s_{i+1}(t-1)$. However, we keep this redundant term in the equations in order to be able to later introduce the same abbreviations for all g_i .

Note, that each of $g_{n-1}, g_{n-2}, \dots, g_a^*$ depends on variables with indexes smaller or equal than b only since, by assumption, the condition (4) holds after the shifting.

A substitution $sub(v_i, v_{i+1})$ is equivalent to replacing the variable $s_i(t-1)$ in the equation of each successor of v_i by $s_{i+1}(t-2)$. After the sequence of a substitutions $sup(v_0, v_1), \dots, sup(v_{a-1}, v_a)$, each $s_i(t-1)$ gets replaced by $s_a(t-1-(a-i))$, so the above equations reduce to:

$$\begin{cases} s_{n-1}(t) = s_a(t-a-1) \oplus g_{n-1}(s_a(t-a-1), s_a(t-a), \dots, s_a(t-1-(a-b))) \\ \dots \\ s_a(t) = s_{a+1}(t-1) \oplus s_a(t-1-a+k)s_a(t-1) \\ \quad \oplus g_a^*(s_a(t-a-1), s_a(t-a), \dots, s_a(t-1-(a-b))) \end{cases}$$

To shorten the expressions, let us introduce an abbreviation $\tilde{s}_a := (s_a(t-a-1), s_a(t-a), \dots, s_a(t-1-(a-b)))$ and let the notation $\tilde{s}_a(i)$ mean that each element $s_a(x)$ of \tilde{s}_a is replaced by $s_a(x+i)$. For example, $\tilde{s}_a(-1) = (s_a(t-a-2), s_a(t-a-1), \dots, s_a(t-2-(a-b)))$. Then, the above equations can be re-written us:

$$\begin{cases} s_{n-1}(t) = s_a(t-a-1) \oplus g_{n-1}(\tilde{s}_a) \\ \dots \\ s_a(t) = s_{a+1}(t-1) \oplus g_a^*(\tilde{s}_a) \oplus s_a(t-1-a+i)s_a(t-1) \end{cases}$$

After a sequence of $n-a-1$ substitutions $sub(v_{n-1}, v_{n-2}), \dots, sub(v_{a+1}, v_a)$, we get a non-linear recurrence describing the sequence of values of the bit a :

$$\begin{aligned} s_a(t) &= s_a(t-n) \oplus g_{n-1}(\tilde{s}_a(-n+a+1)) \oplus g_{n-2}(\tilde{s}_a(-n+a)) \\ &\quad \oplus \dots \oplus g_a^*(\tilde{s}_a) \oplus s_a(t-1-a+i)s_a(t-1) \end{aligned}$$

After expanding the abbreviation \tilde{s}_a , the above recurrence becomes:

$$\begin{aligned} s_a(t) &= s_a(t-n) \\ &\quad \oplus g_{n-1}(s_a(t-n), s_a(t-n+1), \dots, s_a(t-n+b)) \\ &\quad \oplus g_{n-2}(s_a(t-n-1), s_a(t-n), \dots, s_a(t-n+b-1)) \\ &\quad \dots \\ &\quad \oplus g_a^*(s_a(t-a-1), s_a(t-a), \dots, s_a(t-1-a+b)) \\ &\quad \oplus s_a(t-1-a+i)s_a(t-1) \end{aligned} \tag{7}$$

On the other hand, the NLFSR after the shifting can be represented by the following system of equations:

$$\begin{cases} s_{n-1}(t) = s_0(t-1) \oplus g_{n-1}(s_0(t-1), s_1(t-1), \dots, s_b(t-1)) \\ \dots \\ s_a(t) = s_{a+1}(t-1) \oplus g_a(s_0(t-1), s_1(t-1), \dots, s_b(t-1)) \\ s_{a-1}(t) = s_a(t-1) \\ \dots \\ s_b(t) = s_{b+1}(t-1) \oplus s_{i-(a-b)}(t-1)s_b(t-1) \\ \dots \\ s_0(t) = s_1(t-1) \end{cases}$$

After the sequence of b substitutions $sup(v_0, v_1), \dots, sup(v_{b-1}, v_b)$ we get:

$$\begin{cases} s_{n-1}(t) = s_b(t-b-1) \oplus g_{n-1}(s_b(t-b-1), s_1(t-b), \dots, s_b(t-1)) \\ \dots \\ \dots \\ s_a(t) = s_{a+1}(t-1) \oplus g_a^*(s_b(t-b-1), s_1(t-b), \dots, s_b(t-1)) \\ s_{a-1}(t) = s_a(t-1) \\ \dots \\ s_b(t) = s_{b+1}(t-1) \oplus s_b(t-1+i-a)s_b(t-1) \end{cases}$$

Introducing an abbreviation $\tilde{s}_b := (s_b(t-b-1), s_b(t-b), \dots, s_b(t-1))$ we can re-write the above equations us:

$$\begin{cases} s_{n-1}(t) = s_b(t-b-1) \oplus g_{n-1}(\tilde{s}_b) \\ \dots \\ s_a(t) = s_{a+1}(t-1) \oplus g_a^*(\tilde{s}_b) \\ s_{a-1}(t) = s_a(t-1) \\ \dots \\ s_b(t) = s_{b+1}(t-1) \oplus s_b(t-1+i-a)s_b(t-1) \end{cases}$$

After the sequence of $n-b-1$ substitutions $sub(v_{n-1}, v_{n-2}), \dots, sub(v_{b+1}, v_b)$, we get a non-linear recurrence describing the sequence of values of the bit b :

$$\begin{aligned} s_b(t) &= s_b(t-n) \oplus g_{n-1}(\tilde{s}_b(-n+b+1)) \oplus g_{n-2}(\tilde{s}_b(-n+b)) \\ &\oplus \dots \oplus g_b^*(\tilde{s}_b(-(a-b))) \oplus s_b(t-1+i-a)s_b(t-1) \end{aligned}$$

After expanding the abbreviation \tilde{s}_b , the above recurrence becomes:

$$\begin{aligned} s_b(t) &= s_b(t-n) \\ &\oplus g_{n-1}(s_b(t-n), s_b(t-n+1), \dots, s_b(t-n+b)) \\ &\oplus g_{n-2}(s_b(t-n-1), s_b(t-n), \dots, s_b(t-n+b-1)) \\ &\dots \\ &\oplus g_b^*(s_b(t-a-1), s_b(t-a), \dots, s_b(t-1-a+b)) \\ &\oplus s_b(t-1-a+i)s_b(t-1) \end{aligned} \tag{8}$$

The non-linear recurrences (7) and (8) are the same, so two NLFSRs are equivalent.

□